

# Hierarchical Color Correction for Camera Cell Phone Images

Hasib Siddiqui\*

Charles A. Bouman

## Abstract

In this paper, we propose a hierarchical color correction algorithm for enhancing the color of digital images obtained from low quality digital image capture devices such as cell phone cameras. The proposed method is based on a multi layer hierarchical stochastic framework whose parameters are learned in an offline training procedure using the well-known expectation maximization (EM) algorithm. This hierarchical framework functions by first making soft assignments of images into defect classes and then processing the images in each defect class with an optimized algorithm. The hierarchical color correction is performed in three stages. In the first stage, global color attributes of the low quality input image are used in a Gaussian mixture model (GMM) framework to perform a soft classification of the image into  $M$  predefined global image classes. In the second stage, the input image is processed with a non-linear color correction algorithm that is designed for each of the  $M$  global classes. This color correction algorithm, which we refer to as resolution synthesis color correction (RSCC), applies a spatially varying color correction determined by the local color attributes of the input image. In the third stage, the outputs of the RSCC predictors are combined using the global classification weights to yield the color corrected output image. We compare the performance of the proposed method to other commercial color correction algorithms on cell phone camera images obtained from different sources. Both subjective and objective measures of quality indicate that the new color correction algorithm improves quality over the existing methods.

## Index Terms

Color correction, color cast, cell phone camera, resolution synthesis.

## EDICS Categories

SDP-SCAN Scanning and Sampling; ISR-INTR Interpolation; RST-OTHR Other.

The authors are with the School of Electrical and Computer Engineering, Purdue University, 465 Northwestern Ave., West Lafayette, IN 47907; Phone 765-494-0751 and 765-494-0340; FAX (765) 494-3358; E-mail [hsiddiqu@purdue.edu](mailto:hsiddiqu@purdue.edu), [bouman@ecn.purdue.edu](mailto:bouman@ecn.purdue.edu). Corresponding author's e-mail: [hsiddiqu@purdue.edu](mailto:hsiddiqu@purdue.edu).

## I. INTRODUCTION

Digital color enhancement is the process of adjusting the color values in an image so that, on a chosen output device, the processed image looks pleasant to a human viewer. Digital color enhancement is a difficult problem and, depending on the quality of the captured image, may involve various processing steps like contrast enhancement, dynamic range compression, and improving color rendition.

Recently, cell phone cameras have become very popular because of their portability and multipurpose functionality. However, the image quality of existing cell phone cameras is considerably lower than that of most digital still cameras (DSCs). The observed colors in a cell phone camera image are often substantially off from colors in the original scene. Consequently, cell phone camera photos find limited use for archiving printing and sharing.

Most color artifacts in cell phone camera images ensue from poor imaging hardware, including camera optics and image sensors, and subsequent low-cost color processing of the captured data. Typical distortions in cell phone camera photos include poor contrast; incorrect exposure; color fringing and Moiré; color imbalance or global color cast, characterized by a single color dominant in the entire image; and color infidelity, characterized by local shifts in hue that may affect different portions of the captured image differently. The stated artifacts also affect the performance of consumer DSCs to some extent, but are a regular feature in cell phone camera photos.

A variety of different techniques have been proposed in the literature for dynamic range compression, exposure correction, and contrast enhancement in the luminance channel. Some of the proposed techniques include contrast stretching [15], [25], autolevels [30], histogram equalization [14], [15], [22], homomorphic filtering [29], and content-dependent exposure correction [5].

The unwanted global color casts in an image, arising due to changes in illuminant conditions, can potentially be corrected using color constancy processing [10]–[12], [28]. Color constancy algorithms work by first estimating the unknown scene illuminant from an image of the scene. The illuminant estimate is then used to transform the image colors to those relative to a standard illuminant. The technique proposed

in [12] makes use of a trained neural network to recover a two-dimensional estimate of the chromaticity of the ambient illumination given the chromaticity histogram of the test image. Color by correlation [10] works by pre-computing a “correlation matrix”, where the columns of the matrix characterize the possible distribution of image chromaticities under a set of proposed illuminants. Each column of the correlation matrix is then correlated with the chromaticity histogram of the test image to estimate a measure of the likelihood that a specific training illuminant was the scene illuminant. The method proposed in [11] is a fairly recent contribution to the color correction work, and has been demonstrated to perform better than color by correlation and neural network methods. The method makes use of support vector machine based regression for estimating the illuminant chromaticity from the chromaticity histogram of the test image.

Based on Land’s human vision model for lightness and color perception [20], there is also an abundance of Retinex-based image enhancement algorithms [16], [17], [23], including the popular algorithm known as multi scale retinex with color restoration (MSRCR) [17]. MSRCR is a non-linear color correction algorithm whose goal is to improve the overall image quality by providing simultaneous color constancy and contrast enhancement in the luminance channel.

In this paper, we propose a new training-based methodology for color enhancement of low quality cell phone camera images. The novelty of the proposed scheme is that it achieves color enhancement by recovering the minimum mean squared error (MMSE) estimate of a high quality “reference image” given the low quality cell phone camera picture. In our experiments, we use a set of reference images taken with a high quality digital single-lens reflex (SLR) camera and adjusted with Adobe Photoshop. However, different reference images could be used in different applications. In fact, we believe that an important strength of our method is the flexibility to select reference images that reflect the subjective quality attributes that are desired for a specific application.

The proposed algorithm is a non-linear color transformation that can be used to achieve multiple color enhancement objectives; the precise functionality of the algorithm depends on how the training cell phone camera and reference images are selected, and how the algorithm parameters are optimized. Specifically, for discussion in this paper, we will be concerned with correcting the color balance and color infidelity

issues, commonly observed in specific brands of cell phone camera images. The color defects we are trying to fix can arise either due to changes in illuminant, or due to poor imaging hardware and image processing in the cell phone camera.

The proposed algorithm is based on a multi layer hierarchical stochastic framework. The multi layer framework of the algorithm draws inspiration from color by correlation [10] and resolution synthesis (RS) [2], [3]. Similar to the concept in [10], a training procedure is used to learn the parameters for the probability distribution of colors from a set of cell phone camera images displaying a particular type of global color distortion. Next, global color attributes of the test image are used to compute the likelihood that the observed colors in the image are due to each of the global color distortions learned during training. Based on the computed likelihoods, an optimal color transformation is determined for correcting the image. Unlike in [10], the color transformation is non-linear and spatially variant. Using a scheme similar to RS [2], [3], the color transformation at a pixel location is determined by the color of neighboring pixels in the local window. We use pairs of low quality images obtained from a multitude of cell phone camera sources and spatially registered reference images, captured using a high quality digital still camera, to train our algorithm. The resulting pairs of images accurately represent the real-world non-idealities typically found in real mobile camera pictures. We also introduce the use of a generalized Gaussian mixture model (GGMM) [33] for estimating the color corrected output pixel values, and demonstrate that a GGMM can provide better color estimation than a Gaussian mixture model (GMM).

The remainder of this paper is organized as follows. In Sec. II, we introduce the multi layer hierarchical framework of the proposed color correction scheme. Sec. III presents a detailed description of the color correction algorithm. In Sec. IV, we discuss efficient color correction which provides a way to reduce computation and expedite the prediction process. Sec. V contains the experimental results while the concluding remarks are presented in Sec. VI.

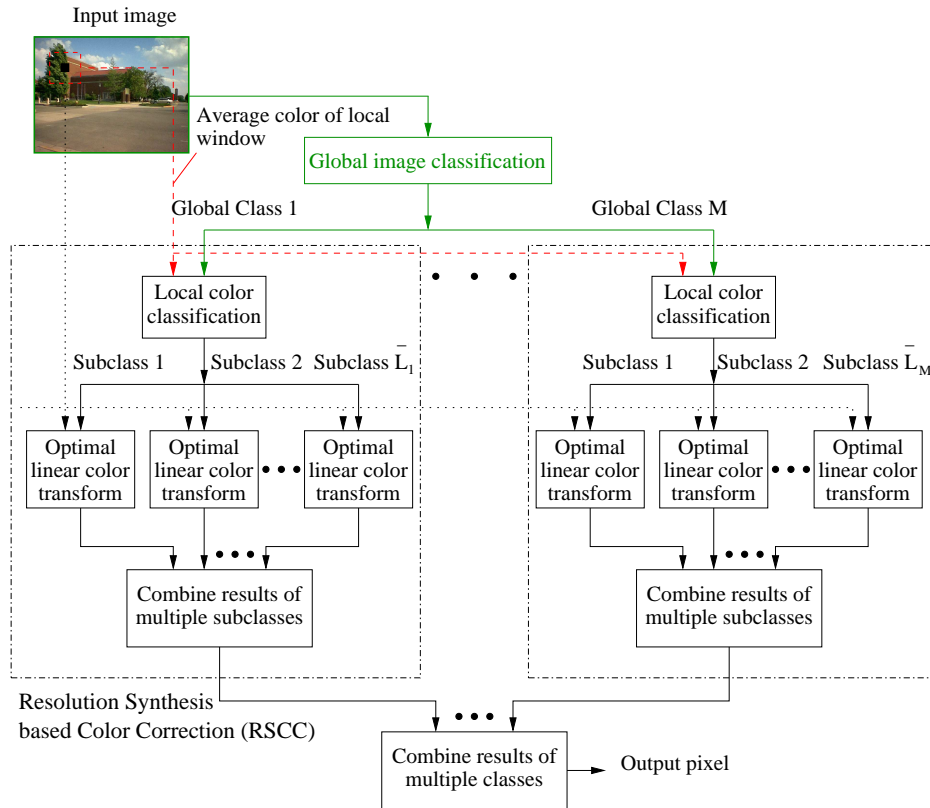


Fig. 1. Multi layer hierarchical structure of color correction. The input image is first classified into multiple classes using the global color classification algorithm, then the average color in a local window is classified into multiple subclasses using a local color classification algorithm. The input pixel is processed with linear color transforms optimized for individual classes and subclasses, and the results of individual color transforms are combined to determine the color of the output pixel.

## II. MULTI LAYER HIERARCHICAL FRAMEWORK FOR COLOR CORRECTION

The proposed color correction scheme uses a two layer hierarchical classifier in order to identify the color distortion at a given pixel location in the input image. Fig. 1 shows how global image classification and local color classification are arranged in a hierarchical framework.

The idea of using global color attributes for determining an optimal color correction for an image has been proposed in the literature. For instance, many machine learning-based color constancy algorithms [10]–[12] make use of the global color histogram of an image to estimate the scene illuminant, based on which an optimal linear color transformation is selected for fixing the color artifacts. The proposed global classification algorithm is fundamentally different from color-constancy processing algorithms in that, instead of estimating the scene illuminant, it classifies the image into different groups, where each

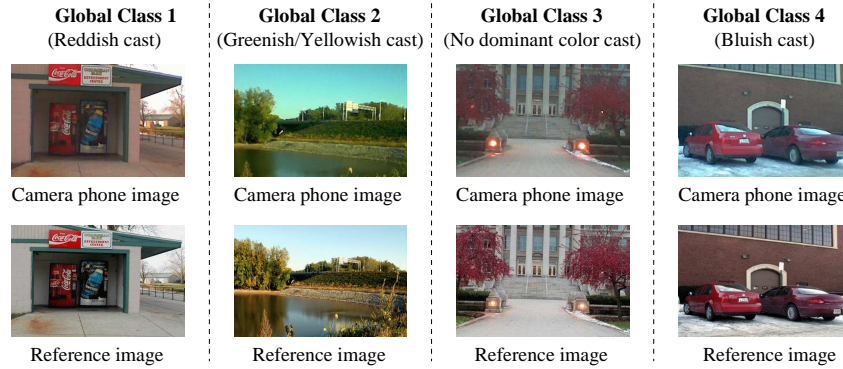


Fig. 2. Typical examples of the global classes assigned to camera phone images used in training.

group characterizes images showing a similar defect behavior. The color defect could arise either due to variation in the scene illumination, or due to sensor metamerism and/or incorrect color processing of the captured data in the low-cost cell phone camera imaging pipeline.

To identify suitable defect classes for our application, we constructed a database comprising pairs of cell phone camera pictures, captured using multiple cell phone sources, and their corresponding ideal representations, herein referred to as the reference images. Details of how the reference images were obtained will be discussed in the following sections. By visually inspecting the cell phone and reference image pairs in our database, we observed that three distinct defect behaviors were most dominant in cell phone camera photos: the cell phone images appeared either too reddish, too bluish, or too greensih/yellowish compared to their corresponding reference images. Moreover, we noticed that images showing a specific defect behavior could be corrected using similar color correction algorithms. Consequently, we came up with the following categories for global image defect classes:

- 1) Global class 1 representing cell phone images with predominantly reddish color cast;
- 2) Global class 2 representing cell phone images with predominantly greenish/yellowish color cast;
- 3) Global class 3 representing cell phone images with no dominant color cast; and
- 4) Global class 4 representing cell phone images with predominantly bluish color cast.

Example cell phone camera and reference image-pairs for the four global classes are shwon in Fig. 2. Thus, for discussion in this paper, we use  $M = 4$ .

The global defect classification algorithm could be based on any one of the many machine learning based

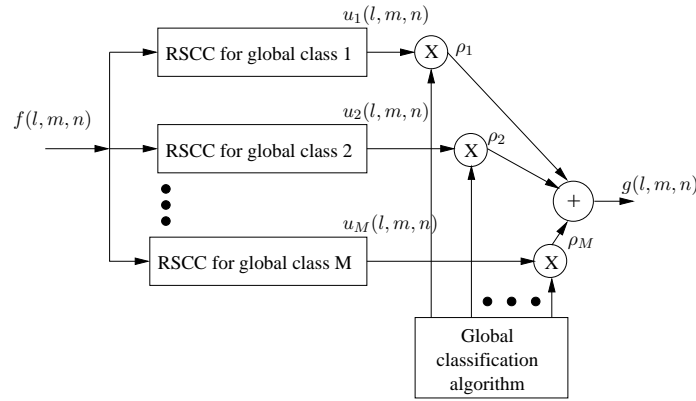


Fig. 3. Hierarchical color correction algorithm. The global classification algorithm computes the classification weights  $\rho_k$ , where  $0 \leq \rho_k \leq 1$  and  $\sum_{k=1}^M \rho_k = 1$ . The outputs of the individual RSCC algorithms are then combined using the global classification weights to determine the final color corrected image.

color constancy algorithms discussed in the literature [10]–[12]. However, we observed that a classification scheme based on a stochastic Gaussian mixture model (GMM) for the image data worked reasonably well for our application.

After the image has been classified into different global defect classes by the global classification algorithm, images from each defect class are processed using a non-linear color correction algorithm which we call resolution synthesis color correction (RSCC). This algorithm is similar to the resolution synthesis predictor discussed in [2], [3], [27] but is adapted for color correction rather than resolution enhancement. The RSCC algorithm extracts the color feature vector from a neighborhood around the current pixel and performs its soft classification into a number of local color subclasses. An affine color transform associated with each subclass is next applied to the current pixel, and then the outputs of the RSCC color transforms are combined to compute the final color corrected image.

While the global classification is based on the traditional Gaussian mixture image model (GMM), we use the heavy-tailed generalized Gaussian mixture image model (GGMM) [33] for local color classification. The heavier tails of the GGMM model result in smoother transitions between colors, thereby minimizing false contouring in the processed image.

### III. HIERARCHICAL COLOR CORRECTION ALGORITHM

Fig. 3 shows a flow diagram of the hierarchical color correction algorithm where  $(m, n)$  indexes the row and column of the image, and the value  $l = 0, 1,$  or  $2$  specifies the red, blue, or green channel of the image, respectively. In all cases, the image is assumed to be represented in the sRGB color space [1] with values in the range 0 to 255.

The cell phone camera image  $f(l, m, n)$  is first input to the global classification algorithm. The global classification algorithm performs a soft classification of the image into  $M$  predefined global classes. The output of the global classification algorithm is a set of  $M$  values  $\rho_k$ , where  $0 \leq \rho_k \leq 1$  and  $\sum_{k=1}^M \rho_k = 1$ . Each  $\rho_k$  gives a measure of the likelihood that the input image belongs to the global class  $k$ .

After global classification, the image  $f(l, m, n)$  is processed with each of the  $M$  RSCC algorithms. Each RSCC predictor is a statistical procedure that applies color correction in order to remove the color defects associated with its global class. Specifically, the goal of RSCC prediction is to recover the MMSE estimate of the high quality reference image given the low quality camera cell phone picture. We denote the output of the  $k$ -th RSCC algorithm by  $u_k(l, m, n)$ . The final image  $g(l, m, n)$  is formed by computing a weighted sum of the outputs of the individual RSCC algorithms:  $g(l, m, n) = \sum_{k=1}^M \rho_k u_k(l, m, n)$ .

#### A. Global Classification Algorithm

To determine the global classification coefficients  $\rho_k$ , the color correction algorithm computes a color feature vector  $\mathbf{y}_{mn}$  at each pixel  $(m, n)$  in the image. The color feature vectors  $\mathbf{y}_{mn}$  are modeled as independent and identically distributed random vectors  $\mathbf{Y}_{mn}$  distributed with an  $M$ -class mixture distribution, where  $M$  is the number of global defect classes used in our color correction framework. We assume that the distribution parameters of the individual mixture model components are denoted by  $\Theta_k$ , where  $1 \leq k \leq M$ . The probability density function of  $\mathbf{Y}_{mn}$  can be written as

$$p(\mathbf{y}_{mn}) = \sum_{k=1}^M p(\mathbf{y}_{mn} | \Theta_k) \pi_k, \quad (1)$$

where  $K \in \{1, \dots, M\}$  is a random variable denoting the global class of the pixel,  $\pi_k$  denotes the *a priori* class probability that the global class of the pixel is  $k$ , and  $p(\mathbf{y}_{mn} | \Theta_k)$  denotes the conditional distribution



of color feature vector.

The log of the probability of the entire data sequence  $\mathbf{y} = \{\mathbf{y}_{mn}\}$  is then given as

$$\log p(\mathbf{y}) = \sum_{m=0}^{M_o-1} \sum_{n=0}^{N_o-1} \log \left[ \sum_{k=1}^M p(\mathbf{y}_{mn} | \Theta_k) \pi_k \right], \quad (2)$$

where  $M_o$  and  $N_o$  are the height and width of the input image, respectively. We assume that the estimates of the class distribution parameters  $\Theta_k$  are already known. The maximum likelihood (ML) estimates of the *a priori* class probabilities  $\pi_k$  can be obtained by maximizing the log likelihood function in (2) iteratively using the expectation maximization (EM) [7], [9], [32] algorithm.

In our global classification algorithm, we run just one iteration of the EM algorithm to obtain the first order estimates of the *a priori* class probabilities,  $\pi_k^{(1)}$ , assuming the initial estimate of the *a priori* class distribution is uniform,  $\pi_k^{(0)} = 1/M$ . The global classification coefficients  $\rho_k$  are then determined as the first order estimates of the *a priori* class probabilities, i.e.,  $\rho_k := \pi_k^{(1)}$ .

Specifically, using the E-step in the EM algorithm, we first compute the posterior class probability for each pixel  $(m, n)$  in the image as

$$p(k | \mathbf{y}_{mn}, \Theta_k) = \frac{p(\mathbf{y}_{mn} | \Theta_k)}{\sum_{k=1}^M p(\mathbf{y}_{mn} | \Theta_k)}. \quad (3)$$

The global classification coefficients  $\rho_k$  are then computed in the M-step of the EM algorithm as

$$\rho_k = \frac{1}{M_o N_o} \sum_{m=0}^{M_o-1} \sum_{n=0}^{N_o-1} p(k | \mathbf{y}_{mn}, \Theta_k). \quad (4)$$

We also model each probability density  $p(\mathbf{y}_{mn} | \Theta_k)$  as a Gaussian mixture. So the combined distribution of (1) becomes a mixture of Gaussian mixtures. The parameters of the Gaussian mixture for defect class  $k$  are given by the parameter vector  $\Theta_k = \left\{ L_k, \left\{ \pi_{j|k}, \boldsymbol{\mu}_{j|k}, \boldsymbol{\Lambda}_{j|k} \right\}_{j=1}^{L_k} \right\}$ , where  $\pi_{j|k}$ ,  $\boldsymbol{\mu}_{j|k}$ , and  $\boldsymbol{\Lambda}_{j|k}^2$  respectively denote the probability, the mean, and the covariance matrix for each component  $j$  in the GMM; and  $L_k$  denotes the number of components in the mixture. The GMM distribution parameter vector  $\Theta_k$  is estimated separately for each class in an offline training procedure using the clustering algorithm in [6]. The training procedure for estimating the parameter vectors  $\Theta_k$  is illustrated in Fig. 4.

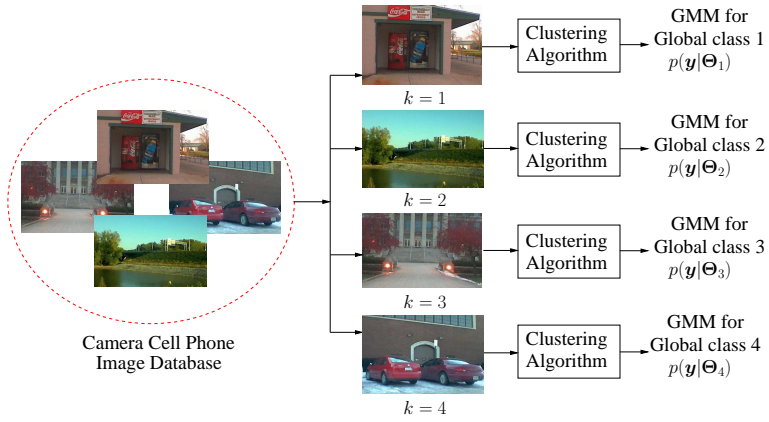


Fig. 4. Training for global classification parameters. The pictures in camera cell phone image database are manually partitioned into  $M$  categories based on their global color properties. The color distribution of images in a global class  $k$  is represented using a Gaussian mixture model  $p(\mathbf{y}|\Theta_k)$ . The mixture distribution parameters  $\Theta_k$  are trained using clustering algorithm [6].

### B. Resolution Synthesis based Color Correction (RSCC)

RSCC is a training-based non-linear color predictor that performs color correction based on the local color in the input image. Fig. 5 shows the structure of an RSCC predictor which is similar to that of the RS predictor developed by Atkins *et al.* [2], [3]. The prediction parameters comprise the classification parameters,  $\bar{\Theta}_k = \left\{ \alpha_k, \bar{L}_k, \left\{ \bar{\pi}_{j|k}, \bar{\mu}_{j|k}, \bar{\Lambda}_{j|k} \right\}_{j=1}^{\bar{L}_k} \right\}$ , and color transform parameters,  $\Psi_k = \left\{ \mathbf{A}_{j|k}, \beta_{j|k} \right\}_{j=1}^{\bar{L}_k}$ , where  $1 \leq k \leq M$ . The notation  $\bar{\Theta}_k$  is used to emphasize that the RSCC classification parameters are different from the global classification parameters  $\Theta_k$  discussed in Sec. III-A. The color transform parameters  $\Psi_k$  comprise  $\mathbf{A}_{j|k}$  and  $\beta_{j|k}$  that represent the affine transform parameters used for each subclass component.

The RSCC prediction parameters are computed in an offline training process discussed in the appendix. For each of the  $M$  RSCC predictors, the training procedure is run independently using a separate set of training data. In each set, the training data comprises pairs of low quality cell phone camera images, showing perceptually similar color distortions, and their corresponding high quality reference images. The training for the RSCC prediction parameters requires very accurate registration of the camera phone and reference images. This means that for each pixel in the cell phone image, the corresponding pixel location in the reference image should be known precisely. The image registration is achieved using the sub-pixel

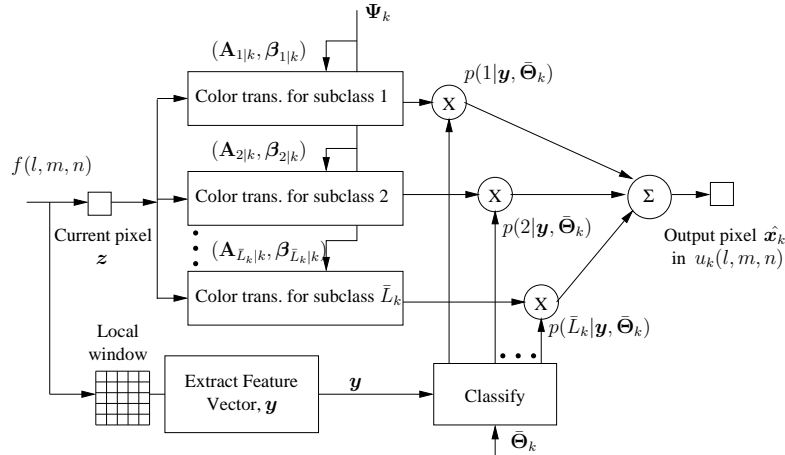


Fig. 5. Illustration of the RSCC predictor. For each global class  $k$ ,  $1 \leq k \leq M$ , there is a unique RSCC predictor with classification parameters  $\bar{\Theta}_k = \left\{ \alpha_k, \bar{L}_k, \left\{ \bar{\pi}_{j|k}, \bar{\mu}_{j|k}, \bar{\Lambda}_{j|k} \right\}_{j=1}^{\bar{L}_k} \right\}$  and color transform parameters  $\Psi_k = \left\{ \mathbf{A}_{j|k}, \beta_{j|k} \right\}_{j=1}^{\bar{L}_k}$ .

registration algorithm developed by Han [13]. Fig. 6 summarizes the process of generating training data for RSCC parameter optimization.

The reference images are captured using Nikon D-100, which is a high quality digital SLR camera. It is common for professional photographers to further enhance images, even when the images have been collected with high quality SLR cameras [21]. In fact, the perceived quality of an image can usually be enhanced by some post processing. For adjusting the colors of our Nikon D-100 images, we typically used “Auto Color” and “Auto Levels” algorithms in Adobe Photoshop. When these algorithms did not work well for a specific image, the image was enhanced manually to represent the best quality in our subjective opinion.

The image  $f(l, m, n)$  is input to each of the  $M$  RSCC predictors. The RGB color value of the input pixel is denoted by the 3-dimensional vector  $z$ . The color feature vector (see Sec. III-B.1) extracted from a local window around the current pixel is denoted by  $y$ . For RSCC prediction, we shall assume  $\mathbf{Y}$  is conditionally distributed as a mixture distribution with parameters  $\bar{\Theta}_k$  given the global defect class  $k$ . Using the parameters  $\bar{\Theta}_k$ , the RSCC classifier computes the probability that the color feature vector  $y$  belongs to a particular subclass  $j$  in the mixture. Using the prediction parameters  $\Psi_k$ , the RSCC predictor next applies optimal affine color transforms for each of the  $\bar{L}_k$  individual subclasses to the current pixel

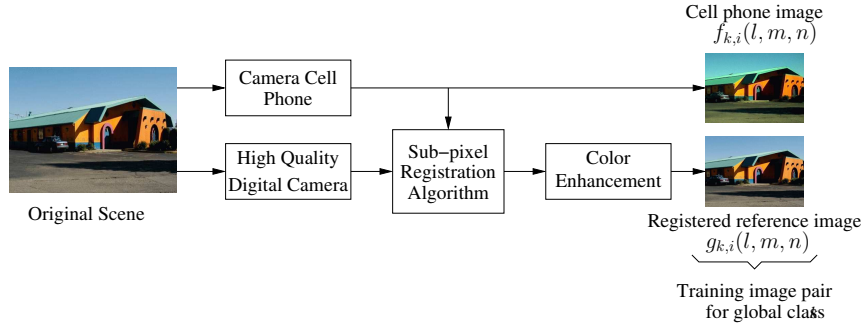


Fig. 6. Generation of a training image pair.  $f_{k,i}(l, m, n)$  is the low quality cell phone image.  $g_{k,i}(l, m, n)$  is the corresponding high quality registered reference image. The training set  $\{f_{k,i}, g_{k,i}\}_{i \in \{1, \dots, I_k\}}$ , where  $I_k$  is the number of images in the training set for the  $k$ -th global image class,  $1 \leq k \leq M$ , is used for training the RSCC parameters  $\bar{\Theta}_k$  and  $\Psi_k$ .

$z$ . The RGB color value of the output pixel  $\hat{x}_k$  is then computed as a weighted linear combination of the outputs of all  $\bar{L}_k$  color transforms with the weighting function for the  $j$ -th color transform corresponding to the probability that the image data is in subclass  $j$ .

While the spatially variant RSCC color transformation improves the accuracy of color correction, it may introduce false edges in visually smooth regions of the image where the neighboring pixels are classified differently due to a small difference in their color values. To smooth the visual contouring, the RSCC predictor uses the heavy-tailed GGMM [33] for modeling the color feature vector instead of the GMM used in the RS predictor. The individual components in a GGMM are represented by generalized Gaussian probability density functions, discussed in Sec. III-B.2.

In the remainder of this section, we proceed as follows. First, we give a description of the color feature vector used for local window classification. Next, we describe the probability density function of a generalized Gaussian random variable and discuss how a heavy-tailed GGMM can help avoid visual contouring. Then we derive the output of the RSCC predictor and discuss its optimality under certain assumptions. The derivation is similar to that given in [2].

1) *Extraction of Color Feature Vector*: The color feature vector  $\mathbf{y}$  is 3-dimensional and contains color information extracted from a  $9 \times 9$  local window in the input image. To obtain the feature vector, we first compute an average of the RGB color values, denoted by  $R_a$ ,  $G_a$ , and  $B_a$ , of similar pixels in the local window. To make the notion of similar pixels more concrete, we perform a color segmentation of

the input image using the algorithm described in [8]. The RGB color averages are then computed using only those pixels in the local window that belong to the same color segment as the center pixel. Next, the average chromaticity values are computed as  $r_a = R_a/(R_a + G_a + B_a)$ ,  $g_a = G_a/(R_a + G_a + B_a)$ , and  $b_a = 1 - r_a - g_a$ . Finally, the feature vector  $\mathbf{y}$  is formed using the following three uncorrelated descriptors of local color:  $Y_a = (0.30R_a + 0.59G_a + 0.11B_a)/255$ ,  $rg_a = r_a - g_a$ , and  $yb_a = (r_a + g_a)/2 - b_a$ .

2) *Generalized Gaussian Probability Density Function*: The probability density function of a generalized Gaussian random variable  $Y$  with mean  $\mu$  and standard deviation  $\sigma$  is given by

$$p(y|\alpha, \mu, \sigma) = \frac{1}{2\sigma A(\alpha)\Gamma(1 + 1/\alpha)} \exp \left[ -\frac{1}{A(\alpha)^\alpha} \left| \frac{y - \mu}{\sigma} \right|^\alpha \right], \quad (5)$$

where  $\Gamma(\cdot)$  is the Euler's gamma function [18],  $A(\cdot)$  is defined as

$$A(x) = \left[ \frac{\Gamma(1/x)}{\Gamma(3/x)} \right]^{1/2}, \quad (6)$$

and  $\alpha > 0$  is a shape parameter that controls the decay rate of the distribution. For  $\alpha = 2$ , (5) reduces to the familiar Gaussian distribution.

Fig. 7 shows plots of generalized Gaussian distributions for the same values of mean  $\mu$  and standard deviation  $\sigma$  but for three different values of  $\alpha$ :  $\alpha < 2$ ,  $\alpha = 2$ , and  $\alpha > 2$ . It can be seen that for  $\alpha < 2$  the distribution is more peaked at the center than the Gaussian density ( $\alpha = 2$ ) but has heavier tails. On the other hand, for  $\alpha > 2$ , the distribution is more flat at the center but shows a sharper roll-off as the distance from the center increases.

Because of heavier tails, the mixture components in a GGMM with  $\alpha < 2$  overlap more extensively than the components in a GMM distribution. The increased overlap among the GGMM components can be exploited to provide a softer classification of the feature vector space. This, in turn, helps avoid the visual contouring associated with non-smooth class transitions.

3) *Optimal Prediction*: In this subsection, the output of RSCC is derived as an MMSE predictor, assuming that the prediction parameters  $\bar{\Theta}_k$  and  $\Psi_k$  are known. The training algorithm for computing the estimates of  $\bar{\Theta}_k$  and  $\Psi_k$  is given in the appendix.

The following three assumptions are made about the image data.

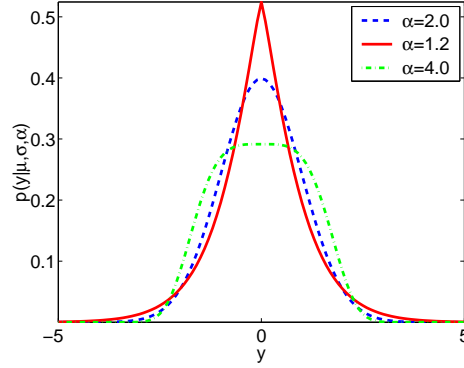


Fig. 7. Generalized Gaussian probability density function ( $\mu = 0.0$ ,  $\sigma = 2.0$ ).

**Assumption 1:** The local color feature vector  $\mathbf{Y}$  is conditionally distributed as a GGMM distribution given the global class  $K$  of the input image. The conditional probability density function  $p(\mathbf{y}|\bar{\Theta}_k)$  is given by

$$p(\mathbf{y}|\bar{\Theta}_k) = \sum_{j=1}^{\bar{L}_k} p(\mathbf{y}|\alpha_k, \bar{\boldsymbol{\mu}}_{j|k}, \bar{\boldsymbol{\Lambda}}_{j|k}) \bar{\pi}_{j|k}, \quad (7)$$

where  $\bar{\Theta}_k$  is the GGMM distribution parameter written as

$$\bar{\Theta}_k = \left\{ \alpha_k, \bar{L}_k, \left\{ \bar{\pi}_{j|k}, \bar{\boldsymbol{\mu}}_{j|k}, \bar{\boldsymbol{\Lambda}}_{j|k} \right\}_{j=1}^{\bar{L}_k} \right\}, \quad (8)$$

where  $\bar{\pi}_{j|k}$  is the probability of subclass  $j$  and  $p(\mathbf{y}|\alpha_k, \bar{\boldsymbol{\mu}}_{j|k}, \bar{\boldsymbol{\Lambda}}_{j|k})$  is the multivariate generalized Gaussian density for subclass  $j$  given by

$$p(\mathbf{y}|\alpha_k, \bar{\boldsymbol{\mu}}_{j|k}, \bar{\boldsymbol{\Lambda}}_{j|k}) = \frac{|\bar{\boldsymbol{\Lambda}}_{j|k}|^{-1}}{[2\Gamma(1 + 1/\alpha_k) A(\alpha_k)]^d} \exp \left\{ - \frac{\left\| \bar{\boldsymbol{\Lambda}}_{j|k}^{-1} (\mathbf{y} - \bar{\boldsymbol{\mu}}_{j|k}) \right\|_{\alpha_k}^{\alpha_k}}{A(\alpha_k)^{\alpha_k}} \right\}, \quad (9)$$

where  $d \in \mathbb{N}$  denotes the dimension of the feature vector  $\mathbf{y}$ ,  $\Gamma(\cdot)$  denotes the Euler's gamma function [18],  $A(\cdot)$  is as defined in (6), and the symbol  $\|\cdot\|_p$  denotes the  $L_p$ -norm.

**Assumption 2:** The conditional distribution of  $\mathbf{X}$  given  $\mathbf{Z}$ ,  $J$ , and  $K$  is multivariate Gaussian, with mean  $\mathbf{A}_{J|K} \mathbf{z} + \boldsymbol{\beta}_{J|K}$ . The RSCC prediction parameters, represented by  $\boldsymbol{\Psi}_k$ , are obtained directly from the parameters for these distributions. So we write  $\boldsymbol{\Psi}_k = \left\{ \mathbf{A}_{j|k}, \boldsymbol{\beta}_{j|k} \right\}_{j=1}^{\bar{L}_k}$ .

**Assumption 3:** The subclass  $J$  is conditionally independent of the vectors  $\mathbf{X}$  and  $\mathbf{Z}$ , given the color feature vector  $\mathbf{Y}$  and the global class  $K$ . Formally, this means that

$$p(j|\mathbf{z}, \mathbf{x}, \bar{\Theta}_k) = p(j|\mathbf{y}, \bar{\Theta}_k). \quad (10)$$

Using these assumptions, for each RSCC predictor, we may compute the MMSE [26] estimate of the high quality pixel given the low quality input pixels and the global defect class as

[26] as

$$\hat{\mathbf{X}}_k = E[\mathbf{X}|\mathbf{Z}, K = k] = \sum_{j=1}^{\bar{L}_k} E[\mathbf{X}|\mathbf{Z}, K = k, J = j] p(j|\mathbf{Z}, \bar{\Theta}_k) \quad (11)$$

$$= \sum_{j=1}^{\bar{L}_k} \left( \mathbf{A}_{j|k} \mathbf{Z} + \boldsymbol{\beta}_{j|k} \right) p(j|\mathbf{Y}, \bar{\Theta}_k). \quad (12)$$

Eq. (12) is obtained by invoking assumptions 2 and 3, and using the fact that  $p(j|\mathbf{Z}, \bar{\Theta}_k) = \int p(j|\mathbf{Z}, \mathbf{x}, \bar{\Theta}_k) p(\mathbf{x}|\mathbf{Z}) d\mathbf{x}$ . The distribution  $p(j|\mathbf{y}, \bar{\Theta}_k)$  can be computed using Bayes' Rule as follows

$$p(j|\mathbf{y}, \bar{\Theta}_k) = \frac{\bar{\pi}_{j|k} |\bar{\Lambda}_{j|k}|^{-1} \exp \left\{ -\frac{1}{A(\alpha_k)^{\alpha_k}} \left\| \bar{\Lambda}_{j|k}^{-1} (\mathbf{y} - \bar{\boldsymbol{\mu}}_{j|k}) \right\|_{\alpha_k}^{\alpha_k} \right\}}{\sum_{m=1}^{\bar{L}_k} \bar{\pi}_{m|k} |\bar{\Lambda}_{m|k}|^{-1} \exp \left\{ -\frac{1}{A(\alpha_k)^{\alpha_k}} \left\| \bar{\Lambda}_{m|k}^{-1} (\mathbf{y} - \bar{\boldsymbol{\mu}}_{m|k}) \right\|_{\alpha_k}^{\alpha_k} \right\}}. \quad (13)$$

The final estimate,  $\hat{\mathbf{X}}$ , for the color corrected pixel is computed by combining the results from all the  $M$  RSCC algorithms.

$$\hat{\mathbf{X}} = \sum_{k=1}^M \rho_k \hat{\mathbf{X}}_k, \quad (14)$$

where the coefficients  $\rho_k$  are as described in Sec. III-A.

#### IV. EFFICIENT COLOR CORRECTION

Two critical parameters that affect the performance of the color correction algorithm are the number of global classes  $M$  and the number of subclasses  $\bar{L}_k$  in each of the  $M$  RSCC predictors. In our application, we use  $M = 4$  and  $\bar{L}_k \approx 30$  for each  $1 \leq k \leq M$ . However, as suggested by Eqs. (12) - (14), this would require an excessive amount of computation. The efficient color correction algorithm is developed in order to reduce computation while preserving the output image quality. The algorithm uses only a subset of global defect classes  $K^* \subset \{1, \dots, M\}$  and a subset of RSCC subclasses  $J_k^* \subset \{1, \dots, \bar{L}_k\}$ , where  $k \in K^*$ , for determining the color correction at a given pixel.

The idea is to include only those classes in  $K^*$  and  $J_k^*$  that best represent the global and local color attributes of the image, respectively. Specifically, the sets  $K^*$  and  $J_k^*$  are defined as follows

$$K^* = \left\{ k \in \{1, \dots, M\} : \rho_k > \frac{\lambda}{M}, \lambda \in (0, 1) \right\} \quad (15)$$

and

$$J_k^* = \left\{ j \in \{1, \dots, \bar{L}_k\} : \frac{p(j|\mathbf{y}, \bar{\Theta}_k)}{p(\tilde{j}|\mathbf{y}, \bar{\Theta}_k)} \geq e^{-\delta^2} \right\}, \quad (16)$$

where  $\delta \geq 0$  and  $\tilde{j}$  is the most likely (ML) subclass in the  $k$ -th RSCC predictor.

The color corrected pixel value,  $\hat{\mathbf{X}}^{eff}$ , using the efficient color correction algorithm is given by

$$\hat{\mathbf{X}}^{eff} = \sum_{k \in K^*} \sum_{j \in J_k^*} \rho_k^* \left( \mathbf{A}_{j|k} \mathbf{Z} + \beta_{j|k} \right) \cdot \frac{\bar{\pi}_{j|k} |\bar{\Lambda}_{j|k}|^{-1} \exp \left\{ -\frac{1}{A(\alpha_k)^{\alpha_k}} \left\| \bar{\Lambda}_{j|k}^{-1} (\mathbf{y} - \bar{\boldsymbol{\mu}}_{j|k}) \right\|_{\alpha_k}^{\alpha_k} \right\}}{\sum_{m \in J_k^*} \bar{\pi}_{m|k} |\bar{\Lambda}_{m|k}|^{-1} \exp \left\{ -\frac{1}{A(\alpha_k)^{\alpha_k}} \left\| \bar{\Lambda}_{m|k}^{-1} (\mathbf{y} - \bar{\boldsymbol{\mu}}_{m|k}) \right\|_{\alpha_k}^{\alpha_k} \right\}}. \quad (17)$$

where the coefficients  $\rho_k^*$  are computed as

$$\rho_k^* = \frac{\rho_k}{\sum_{l \in K^*} \rho_l}. \quad (18)$$

## V. EXPERIMENTAL RESULTS

The experiments were conducted on a set of 220 test images obtained from 6 different camera cell phones: Motorola V600, Sony Ericsson P900, Sony Ericsson T300, Samsung E715, Nokia 3650, and JPhone Sharp J-SH52. The test images included both indoor and outdoor pictures.

The training data for optimizing the algorithm parameters was obtained using one high quality digital still camera, a Nikon D-100, and two low quality camera cell phones, a Nokia 3650 and a Sony Ericsson P900. For each scene, we took three pictures in a row, one with each of the three cameras. The Nikon D-100 images were then subjectively enhanced and spatially registered to their corresponding cell phone camera images to generate the registered reference images.

The training set primarily comprised outdoor images captured at different times of the day, typically a couple of hours after sunrise and half an hour before sunset, and were collected over a span of more than one year. The images included a variety of scenes including buildings, people, animals, sky, foliage, water, snow, color charts, and various paintings. The pictures were taken either on a day when the sky was completely clear, or a day that was completely cloudy. Partly cloudy conditions were avoided. On a partly





Fig. 8. Thumbnails of sample training image pairs for optimizing color correction algorithms for (first row) global class 1, (second row) global class 2, (third row) global class 3, and (fourth row) global class 4. In each training image pair, the left image is a cell phone image while the right image is its spatially registered reference image.

cloudy day, the movement of the clouds made it difficult to ensure similar lighting conditions during different shots of the same scene. The cloud motion also made precise spatial registration of cell phone and reference training image pairs difficult. The thumbnails for some of the registered training image pairs for different global classes are shown in Fig. 8.

The algorithm was rigorously tested only on outdoor images. The test images included a variety of scenes, taken using different cell phone cameras, and were acquired at different times of the day under clear sky to fully cloudy conditions. There were 220 photos in our test image database. None of the test images was used for training the algorithm.

The color correction performance of the proposed algorithm on camera phone images was compared against 4 other methods: “Auto Color” in Adobe<sup>®</sup> PhotoShop<sup>®</sup> 7.0, “Smart White Balance” in Corel<sup>®</sup> Paint Shop Pro<sup>®</sup> X, gray-world white balance [4], and “Retinex” in TruView PhotoFlair<sup>®</sup>. TruView PhotoFlair<sup>®</sup> is a commercial color correction software package based on the Retinex image enhancement algorithm, MSRCR [17].

The color corrected RGB pixel values using the gray world white balance algorithm are determined

$k$	1	2	3	4
$L_k$	50	50	50	50
$L_k^*$	39	35	43	32

$k$	1	2	3	4
$\bar{L}_k$	25	25	30	25
$\bar{L}_k^*$	19	21	28	18
$\alpha_k$	1.2	1.2	1.2	1.2

TABLE I: (A) GMM PARAMETERS IN GLOBAL CLASSIFICATION ALGORITHM. (B) GGMM PARAMETERS FOR DIFFERENT RSCC PREDICTORS.

as:  $R_{GW} = (128/\bar{R}).R$ ,  $G_{GW} = (128/\bar{G}).G$ , and  $B_{GW} = (128/\bar{B}).B$ , where  $\bar{R}$ ,  $\bar{G}$ , and  $\bar{B}$  denote the averages of the three RGB color channels.

The camera phone images were divided into 4 global classes,  $M = 4$ , shown in Fig. 2. Table (a) lists the model orders for the Gaussian mixture distributions used with each of the four global classes. Note that  $L_k$  represents the initial value of the model order, selected manually, for the  $k$ -th global class, while  $L_k^*$  represents the optimized value of the model order estimated from the training images in the  $k$ -th global class. We make use of the penalized Rissanen likelihood criterion for model order estimation [24]. In practice, a higher initial guess,  $L_k$ , tends to produce a higher value of  $L_k^*$ .

The parameters of the generalized Gaussian mixture models associated with each of the four RSCC predictors are listed in Table (b). As before,  $\bar{L}_k$  and  $\bar{L}_k^*$ , respectively, denote the initial number of subclasses, selected manually, and the optimal number of GGMM subclasses, based on the optimality of the penalized Rissanen likelihood criterion. The decay factor of the generalized Gaussian components,  $\alpha_k^*$ , is selected manually based on the observed performance of color correction. Fig. 8 shows how the value of  $\alpha_k$  influences the performance of color correction. We see that, with the selected value of the decay rate ( $\alpha_k = 1.2$ ), the GGMM-based RSCC color predictor avoids false edges due to non-smooth local color classification and provides significantly better color rendition than the GMM-based color predictor ( $\alpha_k = 2.0$ ). Finally, for efficient color correction (see Sec. IV), we select  $\lambda = 0.5$  and  $\delta = 2.2$ .

Figs. 10 - 15(a) show examples of outdoor camera phone images, featuring different kinds of color distortions, obtained from two different sources, Nokia 3650 and Samsung E715. The images in Figs. 10 - 13(a) are obtained from a Nokia 3650 while the images in Figs. 14, 15(a) are obtained from a Samsung

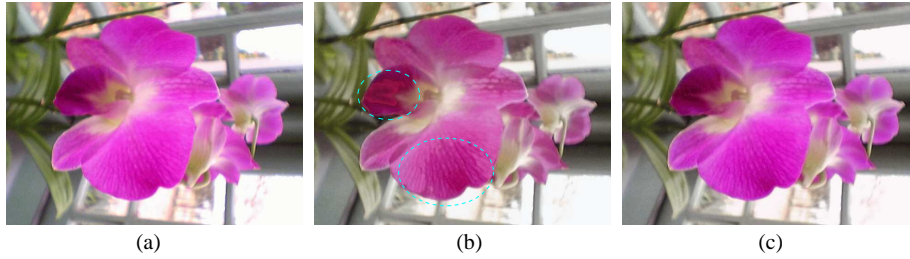


Fig. 9. Illustrating improved color prediction with GGMM. (a) Original image. (b) Processed image with GMM-based RSCC predictors ( $\alpha_k = 2.0$ ). Processed image shows false edges in circled regions. (c) Processed image with GGMM-based RSCC predictors ( $\alpha_k = 1.2$ ).

E715. We recall that Samsung E715 was not used for training the algorithm. While the prediction parameters were trained on the Nokia 3650, none of the test images shown was used during training. Figs. 10 - 15(b) show the corresponding reference images. Figs. 10 - 15(c) show the results of processing the camera phone images with the “Auto Color” algorithm in Adobe<sup>®</sup> PhotoShop<sup>®</sup> 7.0. The results in Figs. 10 - 15(d) are produced using the “Smart White Balance” algorithm in Corel<sup>®</sup> Paint Shop Pro<sup>®</sup> X. The processed images using the white balance algorithm based on the gray world assumption are shown in Figs. 10 - 15(e). The results in Figs. 10 - 15(f) are produced using the “Retinex” algorithm in TruView PhotoFlair<sup>®</sup>. Finally, Figs. 10 - 15(g) show the results from the proposed hierarchical color correction algorithm. We see that the cell phone images processed with the proposed color correction method show the highest fidelity with the high quality reference images.

Fig. 16 shows the performance of the various color correction algorithms on an example indoor image captured using a Nokia 3650. This is particularly interesting because the prediction parameters were trained only on outdoor photos.

The results of global classification for the 7 example images are listed in Table II. Each value of  $\rho_k$  gives a measure of the extent to which the labeled cell phone image (see Figs. 10 - 16) belongs to the global class  $k$ .

In Table III, we use the YCxCz/Lab color image fidelity metric [19], [31] as a quantitative measure of each color correction algorithm’s performance on test images in Figs. 10 - 14. Each entry in the table represents the color difference in terms of the YCxCz/Lab mean squared error measure  $\Delta E$  between

Image no.	$\rho_1$	$\rho_2$	$\rho_3$	$\rho_4$	Dominant class
1	0.511	0.181	0.192	0.116	1
2	0.239	0.418	0.204	0.139	2
3	0.099	0.721	0.072	0.108	2
4	0.270	0.119	0.330	0.201	3
5	0.312	0.235	0.167	0.286	1
6	0.341	0.122	0.166	0.371	4
7	0.327	0.146	0.339	0.188	3

TABLE II: RESULTS OF GLOBAL IMAGE CLASSIFICATION.

Image no.	Unprocessed cell phone image	Processed cell phone image			
		Auto Color	White Balance	MSRCR	Proposed method
1	17.31	17.92	16.36	21.81	13.65
2	17.09	25.25	16.64	21.07	13.24
3	24.24	23.53	19.11	25.00	14.11
4	13.13	18.23	13.47	20.86	10.43
5	19.94	17.72	19.89	25.01	16.60

TABLE III: YCx Cz/LAB VISUALLY WEIGHTED IMAGE FIDELITY METRIC [19], [31]. EACH ENTRY DENOTES THE YCx Cz/LAB MEAN SQUARED ERROR  $\Delta E$  BETWEEN THE LABELED CAMERA PHONE IMAGE AND THE CORRESPONDING REGISTERED HIGH QUALITY REFERENCE IMAGE.

the labeled camera cell phone image and the corresponding high quality registered reference image. The YCx Cz/Lab fidelity metric is based on the color difference in CIE-L\*a\*b\*, but also accounts for the low pass characteristic of the contrast sensitivity of the human visual system.

Table IV compares the average performance of the various color correction algorithms on three different cell phone cameras, a Nokia 3650, a Samsung E715, and a Sony Ericsson P900. In this table, we also compare the performance gain achieved using GGMM ( $\alpha_k = 1.2$ ) for color prediction over GMM ( $\alpha_k = 2.0$ ). The algorithm was trained on the Nokia 3650 and Sony Ericsson P900, but not on Samsung E715. For each cell phone camera, we use a set of 20 image pairs comprising low quality cell phone pictures and registered high quality reference images. The YCx Cz/Lab mean squared error is computed separately for each image pair and then averaged over the entire set to obtain the values listed in the table.

Based on the observed performance on our test image database, which mostly comprised outdoor

Cell phone camera	Unprocessed cell phone image	Processed cell phone image				
		Auto Color	White Balance	MSRCR	RSCC with GMM	RSCC with GGMM
Nokia 3650	17.89	21.73	16.37	22.31	13.55	12.76
Samsung E715	19.94	23.72	18.89	22.71	16.71	15.89
Sony Ericsson P900	18.24	23.44	18.13	23.60	17.03	16.51

TABLE IV: AVERAGE VALUES OF YCx Cz/LAB VISUALLY WEIGHTED IMAGE FIDELITY METRIC [19], [31] FOR DIFFERENT CAMERA CELL PHONES. FOR GMM  $\alpha_k = 2.0$ , WHILE FOR THE GGMM WE USED  $\alpha_k = 1.2$ .

scences, the proposed color correction appears to be robust with respect to change in subject matter and illuminant under outdoor conditions. The color correction performance is also fairly consistent across different cell phone cameras; though, it is important to point out that color correction is more effective for those cell phone cameras whose image samples are used in training the algorithm.

The algorithm was evaluated only on a limited set of indoor pictures. From the observed results, the proposed color enhancement framework appears to also be capable of correcting color balance and color fidelity artifacts in indoor pictures. However, for a robust performance across different indoor illuminants, the algorithm will need to be appropriately trained.

## VI. CONCLUSION

The proposed color correction algorithm, based on global and local classification of image color attributes, provides a robust color correction of a wide range of commercially available camera phone pictures. The training based approach helps the algorithm to learn color distortions that are commonly produced by the low cost mobile cameras. Color correction is then achieved by applying spatially varying color transformations that have been optimized beforehand for recovering the MMSE estimate of the high quality reference image from the low quality input. Both subjective evaluation and a visually weighted image quality metric [19] show that the proposed algorithm performs better on camera cell phone images than a range of other commercially available color correction algorithms.

## VII. ACKNOWLEDGMENTS

This research was funded by Hewlett-Packard Inc. The authors would like to thank Morgan Schramm for his valuable input and comments and Yeesoo Han for his image registration software.

### APPENDIX

#### ESTIMATING PREDICTOR PARAMETERS

In this section, we discuss maximum likelihood estimation [26] for the RSCC classification and color transform parameters, denoted by  $\bar{\Theta}_k$  and  $\Psi_k$  respectively, given the set of training image data. The set of training images is represented as  $\{f_{k,i}(l, m, n), g_{k,i}(l, m, n)\}_{i \in \{1, \dots, I_k\}}$ , where  $f_{k,i}(l, m, n)$  are low quality cell phone images,  $g_{k,i}(l, m, n)$  are high quality registered reference images, and  $I_k$  is the total number of images in the  $k$ -th training set. The set of pixel locations selected for extracting training vectors from training images is denoted by  $S_k$ . The cardinality of  $S_k$  is represented by  $N_k$ . The training vectors include example realizations of the triplet  $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$  drawn from the training images. The extracted training vectors are represented as  $(\mathbf{x}, \mathbf{y}, \mathbf{z}) = (\mathbf{x}_s, \mathbf{y}_s, \mathbf{z}_s)_{s \in S_k}$ .

The RSCC classification parameters  $\bar{\Theta}_k$  that need to be optimized include the model order  $\bar{L}_k$ ; the model decay rate  $\alpha_k$ ; and the subclass information parameter  $\{\bar{\pi}_{j|k}, \bar{\mu}_{j|k}, \bar{\Lambda}_{j|k}\}_{j=1}^{\bar{L}_k}$ . In the following, we shall use the notation  $\bar{\theta}_{j|k} = \{\bar{\pi}_{j|k}, \bar{\mu}_{j|k}, \bar{\Lambda}_{j|k}\}$  and  $\bar{\theta}_k = \{\bar{\theta}_{j|k}\}_{j=1}^{\bar{L}_k}$ . We shall also assume that the matrix  $\bar{\Lambda}_{j|k}$  is diagonal, defined as  $\bar{\Lambda}_{j|k} = \text{diag}[\bar{\sigma}_{j|k,1}, \dots, \bar{\sigma}_{j|k,d}]$ .

The value of  $\alpha_k$  is selected empirically. For a fixed  $\alpha_k$ , the parameters  $\bar{L}_k$  and  $\bar{\theta}_k$  are estimated by minimizing the so called Rissanen minimum description length (MDL) criterion [6], [24]:

$$\begin{aligned}
 MDL(\alpha_k, \bar{L}_k, \bar{\theta}_k) &= -\log p(\mathbf{y}|\alpha_k, \bar{L}_k, \bar{\theta}_k) + \frac{1}{2}P_k \log(N_k d) \\
 &= -\sum_{s \in S_k} \log \left( \sum_{j=1}^{\bar{L}_k} p(\mathbf{y}_s|\alpha_k, \bar{\theta}_{j|k}) \bar{\pi}_{j|k} \right) + \\
 &\quad \frac{1}{2}P_k \log(N_k d), \tag{19}
 \end{aligned}$$

where  $\log p(\mathbf{y}|\bar{\Theta}_k)$  is the log likelihood of the observed sequence  $\mathbf{y} = \{\mathbf{y}_s\}_{s \in S_k}$ ,  $d$  is the dimensionality of  $\mathbf{y}$  ( $d = 3$ ),  $\frac{1}{2}P_k \log(N_k d)$  is a penalty term added to the log likelihood in order to avoid over fitting of

high order models,  $P_k$  is the number of continuously valued real numbers required to specify the parameter  $\bar{\theta}_k$ , and  $N_k d$  is the number of continuously valued real numbers required to specify the data samples  $\mathbf{y}_s$ . For the current application  $P_k = \bar{L}_k(1 + 2d) - 1$ .

Eq. (19) is optimized using the expectation maximization (EM) algorithm [7], [9], [32]. In the following, we outline the algorithm for determining the optimal estimate  $\bar{\Theta}_k^* = \{\alpha_k^*, \bar{L}_k^*, \bar{\theta}_k^*\}$  of the RSCC classification parameters. The complete derivation of the algorithm is similar to that given in [6].

- 1) Set  $\alpha_k = \alpha_k^*$  (chosen empirically).
- 2) Select a large initial value for the number of subclasses  $\bar{L}_k$ .
- 3) Initialize estimate of  $\bar{\theta}_k$ , denoted as  $\bar{\theta}_k^{(\bar{L}_k, 1)}$ . For each  $1 \leq j \leq \bar{L}_k$ , set  $\bar{\pi}_{j|k}^{(\bar{L}_k, 1)}$  to  $\frac{1}{\bar{L}_k}$ , set each cluster mean  $\bar{\mu}_{j|k}^{(\bar{L}_k, 1)}$  to equal one of the feature vectors from the training set, and set  $\bar{\sigma}_{j|k, l}^{(\bar{L}_k, 1)}$ , where  $1 \leq l \leq d$ , as the sample standard deviation of the  $l$ -th element in the feature vector.
- 4) For a fixed  $\bar{L}_k$  and for  $1 \leq j \leq \bar{L}_k$ , use the iterative update equations (20) - (23) to estimate  $N_{j|k}$ ,  $\bar{\pi}_{j|k}$ ,  $\bar{\mu}_{j|k}$ , and  $\bar{\sigma}_{j|k, l}$ :

$$N_{j|k}^{(\bar{L}_k, i+1)} = \sum_{s \in S_k} p(j|\mathbf{y}_s, \alpha_k, \bar{L}_k, \bar{\theta}_k^{(\bar{L}_k, i)}), \quad (20)$$

$$\bar{\pi}_{j|k}^{(\bar{L}_k, i+1)} = \frac{N_{j|k}^{(\bar{L}_k, i+1)}}{N_k}, \quad (21)$$

$$\sum_{s \in S_k} p(j|\mathbf{y}_s, \alpha_k, \bar{L}_k, \bar{\theta}_k^{(\bar{L}_k, i)}) \left\{ \left| y_{s,l} - \bar{\mu}_{j|k, l}^{(\bar{L}_k, i+1)} \right|^{\alpha_k - 1} \text{sign}(y_{s,l} - \bar{\mu}_{j|k, l}^{(\bar{L}_k, i+1)}) \right\} = 0, \quad (22)$$

$$\bar{\sigma}_{j|k, l}^{\alpha_k(\bar{L}_k, i+1)} = \frac{\alpha_k}{A(\alpha_k)^{\alpha_k}} \frac{1}{N_{j|k}^{(\bar{L}_k, i+1)}} \sum_{s \in S_k} p(j|\mathbf{y}_s, \alpha_k, \bar{L}_k, \bar{\theta}_k^{(\bar{L}_k, i)}) \left| y_{s,l} - \bar{\mu}_{j|k, l}^{(\bar{L}_k, i+1)} \right|^{\alpha_k}. \quad (23)$$

Run Eqs. (20) - (23) for  $i_{final}$  number of iterations, when the change in  $MDL(\alpha_k^*, \bar{L}_k, \bar{\theta}_k^{(\bar{L}_k, i)})$  becomes less than  $\bar{\epsilon} = \frac{1}{100} (1 + 2d) \log(N_k d)$ . Note that  $y_{s,l}$  and  $\bar{\mu}_{j|k, l}$  are the  $l$ -th components of the  $d$ -dimensional vectors  $\mathbf{y}_s$  and  $\bar{\mu}_{j|k}$  respectively.

- 5) Record  $\bar{L}_k$ , the parameter  $\bar{\theta}_k^{(\bar{L}_k, i_{final})}$ , and the value  $MDL(\alpha_k^*, \bar{L}_k, \bar{\theta}_k^{(\bar{L}_k, i_{final})})$ .
- 6) If  $\bar{L}_k > 1$ , combine the clusters  $q$  and  $r$  into a single cluster  $(q, r)$ . Choose the clusters  $q$  and  $r$  so

that the following distance function is minimized:

$$d(q, r) = N_k \sum_{l=1}^d \left\{ \bar{\pi}_{q|k}^{(\bar{L}_k, i_{final})} \log \left( \frac{\bar{\sigma}_{(q,r)|k,l}^{(\bar{L}_k)}}{\bar{\sigma}_{q|k,l}^{(\bar{L}_k, i_{final})}} \right) + \bar{\pi}_{r|k}^{(\bar{L}_k, i_{final})} \log \left( \frac{\bar{\sigma}_{(q,r)|k,l}^{(\bar{L}_k)}}{\bar{\sigma}_{r|k,l}^{(\bar{L}_k, i_{final})}} \right) \right\}. \quad (24)$$

Compute the parameters of the combined cluster  $(q, r)$  using Eqs. (25) - (27):

$$\bar{\pi}_{(q,r)|k}^{(\bar{L}_k)} = \bar{\pi}_{q|k}^{(\bar{L}_k, i_{final})} + \bar{\pi}_{r|k}^{(\bar{L}_k, i_{final})}, \quad (25)$$

$$\sum_{s \in S_k} \left| y_{s,l} - \bar{\mu}_{(q,r)|k,l}^{(\bar{L}_k)} \right|^{\alpha_k - 1} \text{sign} \left( y_{s,l} - \bar{\mu}_{(q,r)|k,l}^{(\bar{L}_k)} \right) \cdot \left\{ p \left( q | \mathbf{y}_s, \alpha_k, \bar{L}_k, \bar{\boldsymbol{\theta}}_k^{(\bar{L}_k, i_{final})} \right) + p \left( r | \mathbf{y}_s, \alpha_k, \bar{L}_k, \bar{\boldsymbol{\theta}}_k^{(\bar{L}_k, i_{final})} \right) \right\} = 0, \quad (26)$$

$$\bar{\sigma}_{(q,r)|k,l}^{\alpha_k(\bar{L}_k)} = \frac{\alpha_k}{N_k A(\alpha_k) \left( \bar{\pi}_{q|k}^{(\bar{L}_k, i_{final})} + \bar{\pi}_{r|k}^{(\bar{L}_k, i_{final})} \right)} \sum_{s \in S_k} \left| y_{s,l} - \bar{\mu}_{(q,r)|k,l}^{(\bar{L}_k)} \right|^{\alpha_k} \cdot \left\{ p \left( q | \mathbf{y}_s, \alpha_k, \bar{L}_k, \bar{\boldsymbol{\theta}}_k^{(\bar{L}_k, i_{final})} \right) + p \left( r | \mathbf{y}_s, \alpha_k, \bar{L}_k, \bar{\boldsymbol{\theta}}_k^{(\bar{L}_k, i_{final})} \right) \right\}. \quad (27)$$

Set  $\bar{L}_k \leftarrow \bar{L}_k - 1$ , and go back to step 4.

- 7) Choose the value  $\bar{L}_k^*$  and the corresponding parameter  $\bar{\boldsymbol{\theta}}_k^{(\bar{L}_k^*, i_{final})} = \bar{\boldsymbol{\theta}}_k^*$  for which the minimum value of the MDL criterion is recorded.

After  $\bar{\boldsymbol{\Theta}}_k^*$  has been determined, the optimal estimate  $\boldsymbol{\Psi}_k^* = \{ \mathbf{A}_{j|k}^*, \boldsymbol{\beta}_{j|k}^* \}_{j=1}^{\bar{L}_k^*}$  of the RSCC color transform parameters is computed using the equations given below. The complete derivation of the equations is given in [2]. Defining

$$N_{j|k} = \sum_{s \in S_k} p(j | \mathbf{y}_s, \bar{\boldsymbol{\Theta}}_k^*), \quad (28)$$

$$\mathbf{b}_s = \begin{pmatrix} \mathbf{x}_s \\ \mathbf{z}_s \end{pmatrix}^t, \quad (29)$$

$$\boldsymbol{\nu}_{j|k} = \begin{pmatrix} \boldsymbol{\nu}_{\mathbf{x}|k,j} \\ \boldsymbol{\nu}_{\mathbf{z}|k,j} \end{pmatrix}^t = \frac{1}{N_{j|k}} \sum_{s \in S_k} \mathbf{b}_s p(j | \mathbf{y}_s, \bar{\boldsymbol{\Theta}}_k^*), \quad (30)$$

$$\boldsymbol{\Sigma}_{j|k} = \begin{pmatrix} \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{x}|k,j} & \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{z}|k,j} \\ \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{z}|k,j}^t & \boldsymbol{\Sigma}_{\mathbf{z}\mathbf{z}|k,j} \end{pmatrix} = \frac{1}{N_{j|k}} \sum_{s \in S_k} \mathbf{b}_s \mathbf{b}_s^t p(j | \mathbf{y}_s, \bar{\boldsymbol{\Theta}}_k^*), \quad (31)$$

we have

$$\mathbf{A}_{j|k}^* = \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{z}|k,j} \boldsymbol{\Sigma}_{\mathbf{z}\mathbf{z}|k,j}^{-1}, \quad (32)$$

$$\boldsymbol{\beta}_{j|k}^* = \boldsymbol{\nu}_{\mathbf{x}|k,j} - \boldsymbol{\Sigma}_{\mathbf{x}\mathbf{z}|k,j} \boldsymbol{\Sigma}_{\mathbf{z}\mathbf{z}|k,j}^{-1} \boldsymbol{\nu}_{\mathbf{z}|k,j}. \quad (33)$$



## REFERENCES

- [1] M. Anderson, R. Motta, S. Chandrasekar, and M. Stokes. Proposal for a standard default color space for the internet – sRGB. In *Proc. IS&T/SID 4th Color Imaging Conf.: Color Science, Systems and Applications*, pages 238–246, Scottsdale, Arizona, November 1996.
- [2] C. B. Atkins. *Classification-Based Methods in Optimal Image Interpolation*. PhD thesis, Purdue University, 1998.
- [3] C. B. Atkins, C. A. Bouman, and J. P. Allebach. Optimal image scaling using pixel classification. In *ICIP*, volume 3, pages 864–867, Thessaloniki, Greece, 2001.
- [4] Kobus Barnard, Lindsay Martin, Adam Coath, and Brian Funt. A comparison of computational color constancy algorithms—part II: Experiments with image data. *IEEE Trans. on Image Processing*, 11(9):985–996, September 2002.
- [5] S. Battiato, A. Bosco, A. Castorina, and G. Messina. Automatic image enhancement by content dependent exposure correction. *EURASIP Journal on Applied Signal Processing*, 2004(12):1849–1860, 2004.
- [6] C. A. Bouman. Cluster: An unsupervised algorithm for modeling Gaussian mixtures. Available from <http://www.ece.purdue.edu/~bouman>, April 1997.
- [7] R. A. Boyles. On the convergence of the EM algorithm. *J. Roy. Stat. Society B*, 45(1):47–50, 1983.
- [8] D. Comaniciu and P. Meer. Robust analysis of feature spaces: color image segmentation. In *Proc. of IEEE Computer Soc. Conf. on Computer Vision and Pattern Recognition*, pages 750–755, June 1997.
- [9] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Society B*, 39:1–38, 1977.
- [10] G. D. Finlayson, S. D. Hordley, and P. Hubel. Colour by correlation: a simple, unifying framework for colour constancy. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(11):1209–1221, November 2001.
- [11] B. Funt and W. Xiong. Estimating illumination chromaticity via support vector regression. In *Proc. of Twelfth Color Imaging Conference: Color Science and Engineering Systems and Applications*, pages 47–52, 2004.
- [12] B. V. Funt, V. Cardei, and K. Barnard. Learning color constancy. In *IS&T/SID Fourth Color Imaging Conference*, Scottsdale, November 1996.
- [13] Yeesoo Han. *Sub-Pixel Registration and Image Analysis with Application to Image Comparison*. PhD thesis, Purdue University, 2005.
- [14] R. Hummel. Image enhancement by histogram transformation. *Comp. Graph. Image Process.*, 6(1):184–195, 1977.
- [15] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1989.
- [16] D. J. Jobson, Z. Rahman, and G. A. Woodell. Properties and performance of a center/surround retinex. *IEEE Trans. on Image Processing: Special Issue on Color Processing*, 6:451–462, March 1996.
- [17] D. J. Jobson, Z. Rahman, and G. A. Woodell. A multi-scale retinex for bridging the gap between color images and the human observation of scenes. *IEEE Trans. on Image Processing: Special Issue on Color Processing*, 6(7):965–976, 1997.
- [18] G. B. Thomas Jr. and R. L. Finney. *Calculus and Analytic Geometry*. Addison Wesley, 1996.
- [19] B. Kolpatzik and C. Bouman. Optimal universal color palette design for error diffusion. *Journal of Electronic Imaging*, 4(2):131–143, April 1995.
- [20] E. H. Land. The retinex theory of color vision. *Scientific American.*, pages 108–129, 1977.
- [21] Ken Milburn. *Digital Photography Expert Techniques*. O’Reilly, 2004.
- [22] Stephen M. Pizer, E. Philip Amburn, John D. Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart Ter Haar Romeny,

- and John B. Zimmerman. Adaptive histogram equalization and its variations. *Comput. Vision Graphics and Image Process.*, 39(3):355–368, 1987.
- [23] Z. Rahman, D. Jobson, and G. A. Woodell. Multiscale retinex for color image enhancement. In *Proc. of IEEE Int'l Conf. on Image Proc.*, 1996.
- [24] J. Rissanen. A universal prior for integers and estimation by minimum description length. *Annals of Statistics*, 11(2):417–431, 1983.
- [25] A. Rosenfeld and A. C. Kak. *Digital Picture Processing*. Academic Press, Orlando, FL, 1982.
- [26] L. L. Scharf. *Statistical Signal Processing*. Addison-Wesley, Reading, MA, 1991.
- [27] Hasib Siddiqui and Charles A. Bouman. Training-based descreeing. *IEEE Trans. on Image Processing*, 16(3):789–802, March 2007.
- [28] Rytis Stanikunas, Henrikas Vaitkevicius, and Janus J. Kulikowski. Investigation of color constancy with a neural network. *Neural Network*, 2004.
- [29] T.G. Stockham. Image processing in the context of a visual mode. *Proc. IEEE*, 60(7), 1972.
- [30] Zia ur Rahman, Daniel J. Jobson, Glenn A. Woodell, and Glenn D. Hines. Image enhancement, image quality, and noise. In *Photonic Devices and Algorithms for Computing VII, Proc. SPIE 5907*, 2005.
- [31] W. S. Geisler V. Monga and B. L. Evans. Linear, color-separable, human visual system models for vector error diffusion halftoning. *IEEE Signal Processing Letters*, 10(7), 2003.
- [32] C. F. J. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.
- [33] J. Zhang and J. W. Modestino. A model-fitting approach to cluster validation with application to stochastic model-based image segmentation. *IEEE Trans. on Image Processing*, 12(10):1009–1017, October 1990.



Fig. 10. Image-1, outdoor scene, classified as predominantly reddish. (a) Camera phone image (Nokia 3650). (b) Reference Image (Nikon D-100). Processed images using (c) “Auto Color” in Adobe PhotoShop<sup>®</sup> 7.0, (d) “Smart White Balance” in Corel Paint Shop Pro<sup>®</sup> X, (e) Gray World White Balance, (f) “Retinex” in TruView PhotoFlair<sup>®</sup>, and (g) Proposed Algorithm.



Fig. 11. Image-2, outdoor scene, classified as predominantly yellowish/greenish. (a) Camera phone image (Nokia 3650). (b) Reference Image (Nikon D-100). Processed images using (c) “Auto Color” in Adobe PhotoShop<sup>®</sup> 7.0, (d) “Smart White Balance” in Corel Paint Shop Pro<sup>®</sup> X, (e) Gray World White Balance, (f) “Retinex” in TruView PhotoFlair<sup>®</sup>, and (g) Proposed Algorithm.



Fig. 12. Image-3, outdoor scene, classified as predominantly yellowish/greenish. (a) Camera phone image (Nokia 3650). (b) Reference Image (Nikon D-100). Processed images using (c) “Auto Color” in Adobe PhotoShop<sup>®</sup> 7.0, (d) “Smart White Balance” in Corel Paint Shop Pro<sup>®</sup> X, (e) Gray World White Balance, (f) “Retinex” in TruView PhotoFlair<sup>®</sup>, and (g) Proposed Algorithm.



Fig. 13. Image-4, outdoor scene, classified as having no dominant color cast. (a) Camera phone image (Nokia 3650). (b) Reference Image (Nikon D-100). Processed images using (c) “Auto Color” in Adobe PhotoShop<sup>®</sup> 7.0, (d) “Smart White Balance” in Corel Paint Shop Pro<sup>®</sup> X, (e) Gray World White Balance, (f) “Retinex” in TruView PhotoFlair<sup>®</sup>, and (g) Proposed Algorithm.



Fig. 14. Image-5, outdoor scene, classified as predominantly reddish. (a) Camera phone image (Samsung E715). (b) Reference Image (Nikon D-100). Processed images using (c) “Auto Color” in Adobe PhotoShop<sup>®</sup> 7.0, (d) “Smart White Balance” in Corel Paint Shop Pro<sup>®</sup> X, (e) Gray World White Balance, (f) “Retinex” in TruView PhotoFlair<sup>®</sup>, and (g) Proposed Algorithm.

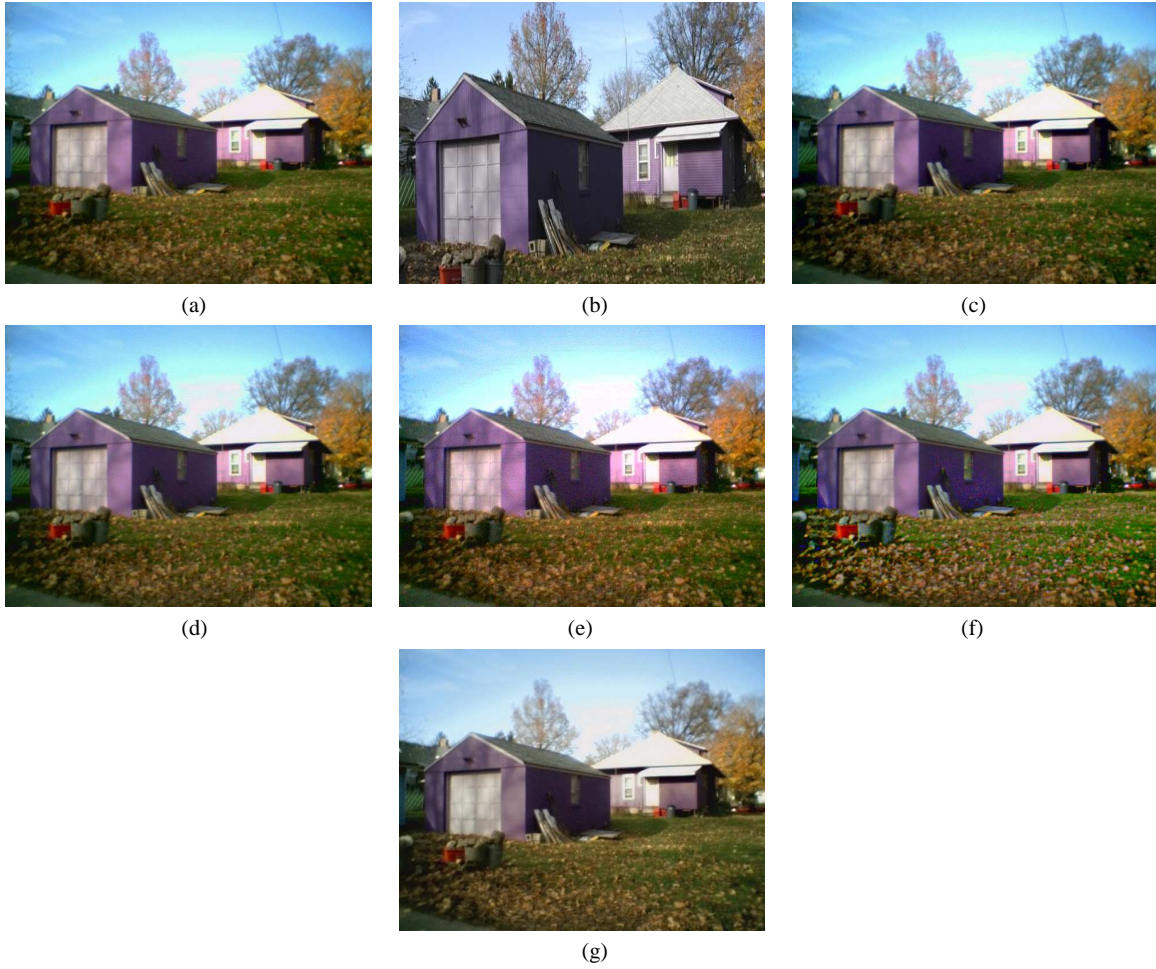


Fig. 15. Image-6, outdoor scene, classified as predominantly bluish. (a) Camera phone image (Samsung E715). (b) Reference Image (Nikon D-100). Processed images using (c) “Auto Color” in Adobe PhotoShop<sup>®</sup> 7.0, (d) “Smart White Balance” in Corel Paint Shop Pro<sup>®</sup> X, (e) Gray World White Balance, (f) “Retinex” in TruView PhotoFlair<sup>®</sup>, and (g) Proposed Algorithm.





(a)



(b)



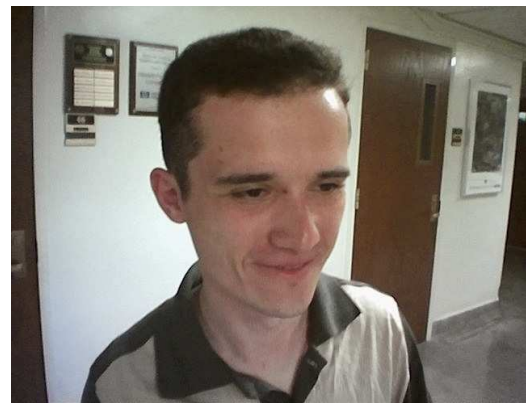
(c)



(d)



(e)



(f)

Fig. 16. Image-7, indoor picture, classified as having no dominant color cast. (a) Camera phone image (Nokia 3650). Processed images using (b) “Auto Color” in Adobe PhotoShop<sup>®</sup> 7.0, (c) “Smart White Balance” in Corel Paint Shop Pro<sup>®</sup> X, (d) Gray World White Balance, (e) “Retinex” in TruView PhotoFlair<sup>®</sup>, and (f) Proposed Algorithm.